Transcrição

Iniciaremos o projeto deste curso de fato como estatísticos ou cientistas de dados, onde faremos a **análise descritiva** do conjunto de dados que já vin

Na seção de **Distribuição de Frequências** do notebook `Curso_de_Estatística_Parte_1.ipynb`, começaremos com uma técnica de **sumarização** p variáveis se distribuem, vendo se são assimétricas ou se são distribuídas normalmente, se conseguiremos detectar *outlier*, ou se precisam de algum ti exemplo.

São **técnicas estatísticas** que lidam com um conjunto de variáveis e buscam conhecer seus **comportamentos**, passando primeiro por formas gráficas auxiliam na análise da Distribuição de Frequências nos dados.

Nesta mesma parte, iniciaremos com as **qualitativas** que são naturalmente categorizadas, então não precisaremos nos preocupar com isso. Como vin as `Sexo`, `UF` e `Cor`.

Na primeira célula, chamaremos os `dados[]` de `'Sexo'` e faremos a **contagem** com o método `.value_counts()`. Executando este comando, vere ocorrências para cada categoria desta variável.

```
dados['Sexo'].value_counts()
```

O retorno da execução mostrará `53250` para a categoria `0` relativa ao sexo masculino, e `23590` para o feminino. Sabemos que isso não corresponc da população brasileira é composta por mulheres, porém a coleta dessas informações foi baseada em somente registros das **pessoas de referência** de responsável pela casa que respondeu o questionário da pesquisa, e neste caso a maioria das respostas foi dada por homens, resultando nos valores qu

Outra informação importante que costuma aparecer nas tabelas de frequências é este mesmo retorno representado de forma **percentual**. Para fazerm método `.value_counts()` recebendo o parâmetro `normalize` sendo igual a `True`.

Isto normalizará os dados e os colocarão na base 1. Se ainda multiplicarmos por `100`, veremos a **porcentagem** de aproximadamente 70% de pessoa do feminino que responderam ao questionário.

```
dados['Sexo'].value_counts(normalize = True) * 100
```

Para melhorarmos a apresentação dos dados, inseriremos o primeiro comando somente com `.value_counts()` sem parâmetro dentro de uma variáv

```
frequencia = dados['Sexo'].value_counts()
```

Feito isso, teremos uma *series* do Pandas. Faremos a mesma coisa com o comando que apresenta o percentual, e o colocaremos dentro da variável `p` seguinte.

```
percentual = dados['Sexo'].value_counts(normalize = True) * 100
```

Para organizarmos melhor a apresentação, colocaremos tudo isso dentro de um novo `DataFrame` que representará a **Tabela de Frequências** a ser ap

Então, chamaremos de **`dist_freq_qualitativas`** sendo igual a `pd.DataFrame()` para o criarmos, onde passaremos um dicionário Python com `{}` duas *series* criadas anteriormente. A primeira coluna será a `'Frequência'` como se fosse um arquivo `.json`, e então adicionaremos a variável `fre`

Após a vírgula, colocaremos a segunda coluna `'Porcentagem (%)'` visto que já multiplicamos por `100`, então não precisaremos formatar o número passaremos a variável `percentual` após `:` de novo.

```
dist_freq_qualitativas = pd.DataFrame({'Frequência': frequencia, 'Porcentagem (%)': percentual})
```

Com isso, poderemos apenas escrever `dist_freq_qualitativas` e executar na célula seguinte para vermos a tabela com as frequências de porcenta

	Frequência	Porcentagem (%)
0	53250	69.299844
1	23590	30.700156

Como vimos anteriormente, a codificação `0` e `1` da variável `Sexo` correspondem às respostas "masculino" e "feminino" do questionário. Então de que cada código significa.

Portanto, atribuiremos *labels* ou **etiquetas**. Na célula seguinte, chamaremos `dist_freq_qualitativas` com **`.rename()`** para renomearmos o `inde` também, dizendo que `0` corresponde a `'Masculino'` e `1` a `'Feminino'`.

```
dist_freq_qualitativas.rename(index = {0: 'Masculino', 1: 'Feminino'})
```

	Frequência	Porcentagem (%)
Masculino	53250	69.299844
Feminino	23590	30.700156

Com a execução, veremos os nomes das etiquetas corretamente. Mas se apenas chamarmos o `DataFrame` de novo na célula seguinte, a alteração nã

Para sobrescrevermos e salvarmos as mudanças no arquivo, deveremos inserir o parâmetro **`inplace`** sendo igual a `True` no comando anterior.

```
dist_freq_qualitativas.rename(index = {0: 'Masculino', 1: 'Feminino'}, inplace = True)
```

Feito isso, o sistema criará e salvará o dicionário. Logo, poderemos apenas chamar a `dist_freq_qualitativas` e executá-la na célula para visualiza códigos de `Sexo`.

Também é possível adicionarmos um título para a tabela com o `DataFrame` seguido de **`.rename_axis()`**, onde passaremos o nome `'Sexo'` que qu labels.

Como podemos renomear tanto uma coluna quanto uma linha, deveremos indicar qual é o eixo **`axis`** que queremos. Neste caso, o Pandas reconhec e `0` como linha, mas para não nos confundirmos, escreveremos `'columns'` ou `'rows'` que funcionam da mesma forma.

Não poderemos esquecer de colocar `inplace` igual a `True` para salvarmos as alterações.

```
dist_freq_qualitativas.rename(index = {0: 'Masculino', 1: 'Feminino'}, inplace = True)
dist_freq_qualitativas.rename_axis('Sexo', axis = 'columns', inplace = True)
```

Executaremos estes comandos e chamaremos apenas `dist_freq_qualitativas` na célula seguinte para visualizarmos os resultados.

Sexo	Frequência	Porcentagem (%)
Masculino	53250	69.299844
Feminino	23590	30.700156

Com isso, teremos uma Tabela de Frequências nomeada com clareza que poderemos copiar, colar e apresentar que será bem entendida. É interessant exercícios com as outras variáveis qualitativas que temos para aprimorarmos as habilidades.

A seguir, veremos uma nova maneira com outro método do Pandas para fazermos esta mesma ação.